**Case Study : Stock Market Prediction**

**Problem Statement:**

* The stock market is a complex and dynamic environment influenced by multiple factors such as company performance, economic conditions, and investor behavior.
* Predicting stock market trends is difficult due to continuous fluctuations and uncertainty. Traditional methods often fail to adapt to new patterns.
* This case study explores how **Artificial Intelligence (AI)** techniques specifically **basic and heuristic search methods** can be applied to predict stock price movement effectively.

**Aim:**

To implement and compare **basic** and **heuristic search techniques** for predicting future stock prices using historical market data.

**Objectives:**

* To study and understand the principles of basic and heuristic search techniques.
* To apply these techniques in the context of stock price prediction.
* To analyze historical stock data and identify potential future trends.
* To compare the performance and efficiency of both search methods.
* To interpret prediction results for decision-making in stock investments.

**Description:**

* This case study focuses on predicting the **future closing price of a stock** based on its **historical data**.
* The data includes parameters such as **Open, High, Low, Close, and Volume**.
* The project uses **machine learning techniques** specifically **Linear Regression** to find the relationship between past and future prices.
* Libraries like yfinance, pandas, numpy, and sklearn are used for data fetching, processing, and modeling.
* By analyzing historical data, the model can make an approximate prediction for the next-day stock price, helping investors make informed decisions.

**Algorithm:**

1. Import necessary Python libraries.
2. Fetch historical stock market data (e.g., Apple stock from Yahoo Finance).
3. Preprocess data and create a new column for the next-day closing price.
4. Split data into training and testing sets.
5. Train the Linear Regression model using training data.
6. Predict the stock’s next-day closing price.
7. Evaluate the model using metrics like MAE or RMSE.
8. Display predicted results.

**Program:**

import yfinance as yf

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error

# Step 1: Fetch stock data

data = yf.download("AAPL", start="2020-01-01", end="2025-01-01")

# Step 2: Prepare data

data["Next\_Close"] = data["Close"].shift(-1)

X = data[["Open", "High", "Low", "Close", "Volume"]][:-1]

y = data["Next\_Close"][:-1]

# Step 3: Split data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Step 4: Train model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Step 5: Predict

predictions = model.predict(X\_test)

# Step 6: Evaluate

error = mean\_absolute\_error(y\_test, predictions)

print("Mean Absolute Error:", error)

print("Predicted Next-Day Price:", predictions[-1])

**Output:**

Mean Absolute Error: 2.45

Predicted Next-Day Price: 225.78 USD

**Conclusion:**

From this case study, it is observed that **machine learning can effectively predict stock prices** based on historical data trends. Although the model provides approximate results, it helps identify **general price movement direction**. However, due to the market’s unpredictable nature, predictions should be used as **supportive tools**, not absolute guarantees. Further improvement can be achieved by using **advanced models** such as **LSTM (Long Short-Term Memory)** or by incorporating **news sentiment analysis**.